มุม dBASE

ยืน ภู่วรวรรณ

ความสัมพันธ์ระหว่างไฟล์และการตั้งกระบวนความ

มุมดีเบสฉบับนี้จะกล่าวถึงขีดความ
มารถของดีเบสทรี ในเรื่องที่สำคัญสอง
อง คือเรื่อง **ความสัมพันธ์ระหว่างไฟล์**
ะ **การตั้งกระบวนความ** ทั้งสองเรื่องนี้
นสิ่งที่ดีเบสทรี พอจะกล่าวอ้างได้ว่า เป็น
เด่นที่สำคัญของการใช้งาน เพราะเรื่อง
องความสัมพันธ์ระหว่างไฟล์ จะทำให้การ
การฐานข้อมูลแบบรีเลชัน เป็นไปได้ง่าย
ะมีประสิทธิภาพ ส่วนการตั้งกระบวน
าม(procedure) ก็เป็นหลักการของภาษา
มือนเช่นของภาษาปาสคาล ซี ฯลฯ นับว่า
เด่นทั้งสองนี้ควรแก่การศึกษา และหาทาง
ไปใช้เป็นอย่างยิ่ง ลองดูเถอะครับ แล้วคุณ
เห็นเองว่าดีเบสทรีน่าใช้เพียงไร

วามสัมพันธ์ระหว่างไฟล์

ก่อนที่จะมาดูการใช้งานแฟ้มข้อมูลจริง
มาดูหลักการของดีเบสก่อน ดีเบสตั้งพื้นที่
หน่วยความจำสำหรับประมวลผลแฟ้ม
มูลได้ ๑๐ แฟ้ม โดยใช้คำสั่ง SELECT 1
SELECT 10 เช่นถ้าใช้

```
.SELECT 1
.USE NAMECARD
```

ก็หมายถึงนำแฟ้มข้อมูล NAME-CARD จากฐานข้อมูลมายังพื้นที่ทำงาน (ซึ่งหมายถึงนำจากดิสค์มายังหน่วยความจำ) โดยผู้ใช้สามารถตั้งชื่อเล่นเป็น ALIAS NAME ได้ เช่น

```
.SELECT 1
.USE NAMECARD ALIAS CARD
```

การตั้งชื่อ ALIAS นี้ คือชื่อพื้นที่ ซึ่งก็หมายถึงพื้นที่ที่ 1 ดังนั้นถ้าคราวต่อไปจะอ้างถึงพื้นที่ที่ 1 ก็จะใช้

```
.SELECT CARD
```

การเลือก ALIAS แทนหมายเลขนี้เพื่อกำหนดชื่อ และเป็นสิ่งที่เข้าใจได้ง่าย แต่ใครจะใช้ SELECT 1 อย่างเดิมก็ได้ แต่ถ้าจะอ้างข้อมูลใน ALIAS อื่น ต้องใส่ชื่อ ALIAS เสมอ ถ้าไม่ใส่ดีเบสจะกำหนดเป็น A, B, C..... ตามหมายเลข ALIAS คือ 1 แทนด้วย A

การจัดการข้อมูลหลายแฟ้ม

สมมติห้องสมุดแห่งหนึ่งต้องการบันทึกผู้ยืมหนังสือ จึงทำบัตรประจำตัวสมาชิกห้องสมุด ซึ่งลักษณะบัตรจะเป็นดังนี้

Micro Lab

IDNO.....

NAME.....

ADDRESS.....

TELEPHONE.....

วันที่ยืม	ชื่อหนังสือ	สถานะส่งคืน

dBASE

ปัญหาของระบบข้อมูลคือ เราไม่สามารถจัดแฟ้มข้อมูลแบบตาราง เช่นดีเบสในไฟล์เดียวได้หมด เพราะโครงสร้างข้อมูลเป็นแบบลำดับขั้น กล่าวคือถ้าเราจะเขียนไฟล์เพียงไฟล์เดียว จะไม่สามารถกำหนดข้อมูลที่เสมือนเป็นตารางซ้อนได้อีก และในสภาพของการใช้งานจริง เราจึงมักพบข้อมูลแบบตารางซ้อนตารางหลายๆ ชั้น การจัดโครงสร้างแฟ้มข้อมูลของเราจึงจัดได้ โดยแบ่งแยกข้อมูลออกเป็นหลายแฟ้มข้อมูล โดยให้แต่ละแฟ้มมีส่วนเชื่อมโยงถึงกัน

ในที่นี้เราแบ่งแยกออกเป็นแฟ้ม NAMECARD และ BORROW ดังรูปที่ 1

แฟ้มข้อมูล NAMECARD นี้ จะมีเพียงรหัส IDNO, NAME, ADDRESS และ TELEPHONE ซึ่งจะต้องเชื่อมโยงกับอีกแฟ้มหนึ่งคือแฟ้ม BORROW ซึ่งมีโครงสร้างดังรูปที่ 2

สังเกตข้อมูลที่ใส่ไว้ในแฟ้มข้อมูลนี้ มีตัวเชื่อมโยงคือ IDNO และจากข้อมูลทั้งสองแฟ้มบอกกับเราได้ว่า srisuda ยืมหนังสือของห้องสมุดไป 2 รายการ ซึ่งจะมีรายกา
ทั้ง 2 ไปปรากฏในแฟ้ม BORROW ดังนั้
ในการ์ดของผู้ยืมถ้ามี 10 รายการ ก็หมายถึ
มีข้อมูลอยู่ในแฟ้ม BORROW 10 เรคอร์

การ set relation

ดีเบสทรีให้ผู้ใช้ตั้งความสัมพันธ์ขอ
แฟ้มข้อมูลได้โดยตรง เพื่อให้เกิดความเข้า
ได้ง่าย จะเขียนไดอะแกรมของแฟ้มข้อมู
ทั้งสองเป็นตารางดังรูปที่ 3

1

```
. USE NAMECARD
. LIST STRU
Structure for database : b:NAMECARD.dbf
Number of data records :        4
Date of last update    : 01/01/80
Field  Field name  Type        Width     Dec
    1  IDNO        Character       4
    2  NAME        Character      20
    3  ADDRESS     Character      20
    4  TELEPHONE   Character       7
** Total **                      52

. LIST
Record#  IDNO NAME                 ADDRESS              TELEPHONE
      1  A001 srisuda varaporn     1 new road Bkk.      2324322
      2  A002 johny walker         2 sukhumvit 29 Bkk.  3424566
      3  A003 dangdum seesod       2 vipavadee Bkk.     5438765
      4  A004 ngamta sopapan       3 paholyothin        5792534
```

2

```
  USE BORROW
  LIST STRU
  ucture for database : b:BORROW.dbf
Number of data records :        9
Date of last update    : 01/01/80
Field  Field name  Type        Width     Dec
    1  IDNO        Character       4
    2  BOOKNAME    Character      15
    3  DATEBORROW  Date            8
    4  STATUS      Logical         1
** Total **                      29

. LIST
Record#  IDNO BOOKNAME        DATEBORROW STATUS
      1  A001 dBASE III       03/21/86   .F.
      2  A001 lotus 123       04/11/86   .F.
      3  A002 Thaieasy ver4.5 04/08/86   .F.
      4  A002 lotus release 2 04/25/86   .T.
      5  A002 C programming   02/26/86   .F.
      6  A003 8086 Assembly   04/17/86   .T.
      7  A004 Local area net. 04/25/86   .F.
      8  A004 Thaieasy ver4.1 02/24/86   .F.
      9  A004 MSDOS           04/21/86   .T.
```

ความสัมพันธ์ระหว่างไฟล์นี้ จะเสมือ
เครื่องเชื่อมโยงให้แฟ้มทั้งสองแฟ้ม เสมือ
เป็นแฟ้มเดียวกัน และใช้งานได้คล่องตัว
ลองดูการใช้ในดีเบสทรีดู

เราเริ่มจากการทำอินเด็กซ์ของไ
BORROW โดยใช้คีย์ฟิลด์ IDNO เพร
จะต้องเชื่อมโยงกันด้วย IDNO ดังรูปที่

ในกรณีที่เราเคย index มาแล้ว ดีเ
จะถามว่าจะเขียนทับหรือไม่

จากนั้นเราจะเริ่มทดลองการ Set re
tion โดยรูปแบบของคำสั่งเป็น

Set relation to<ชื่อฟิลด์>INTO<ชื่อ ALIAS

ขั้นตอนจึงเป็นดังรูปที่ 5

การเริ่มต้นจาก select 1 แล้วใช้ไ
NAMECARD และตั้งชื่อ ALIAS เป็

3

relation

IDNO	NAME	ADDRESS	TEL
A001	srisuda	1 new road	2324322
A002	johny	2 Sukhumvit	3424566
A003	dangdum	2 vipavadee	5438765
A004	ngamta	3 paholyothin	5792534

	IDNO	BOOKBORROW	DATE	STATUS
1	A001	dBASE III	03/21/86	F
2	A001	lotus 123	04/11/86	F
3	A002	Thaieasy ver4.5	04/08/86	F
4	A002	lotus	04/25/86	T
5	A002	C programming	02/26/86	F
6	A003	8086 Assembly	04/17/86	T
7	A004	Local area	04/28/86	F
8	A004	Thaieasy	02/24/86	F
9	A004	MSDOS	04/21/86	T

4

```
.
. USE BORROW
. INDEX ON IDNO TO BORROW
BORROW.ndx already exists, overwrite it? (Y/N) Yes
      9 records indexed
```

5

```
. SELECT 1
. USE NAMECARD ALIAS CARD
. SELECT 4
. USE BORROW INDEX BORROW ALIAS BORR
. SELECT CARD
. SET RELATION TO IDNO INTO BORR
.
. GOTO 3
. ? RECNO()
        3
. DISPLAY
Record#  IDNO NAME                    ADDRESS              TELEPHONE
      3  A003 dangdum seesod          2 vipavadee Bkk.     5438765
. SELECT BORR
. DISPLAY
Record#  IDNO BOOKNAME        DATEBORROW STATUS
      6  A003 8086 Assembly   04/17/86   .T.
.
. SELECT CARD
. GOTO 2
. ? IDNO,NAME,ADDRESS,BORR->BOOKNAME,BORR->DATEBORROW,BORR->STATUS
A002 johny walker          2 sukhumvit 29 Bkk.  Thaieasy ver4.5 04/08/86 .F.
```

ARD การตั้งชื่อ ALIAS จะตั้งชื่อเหมือน ไฟล์ไม่ได้ คำสั่ง set relation to idno o BORR จึงหมายถึง ให้นำไฟล์ที่กำลัง ใน ALIAS CARD ไปสัมพันธ์กับไฟล์ ALIAS borr โดยให้ IDNO เป็นตัวเชื่อม

ดังนั้นเมื่อเลื่อน Recno( ) ไปที่ 3 ใน IAS CARD จะทำให้ Recno ( ) ใน IAS borr เลื่อนไปเรคอร์ด 6 ซึ่งตรง พอดี

ด้วยหลักการนี้ เราจึงนำไปเขียนโปร-แกรมได้โดยง่ายขึ้นกว่าเดิม เพราะเราไม่ต้องใช้คำสั่ง FIND ในการค้นหาข้อมูลในต่าง ALIAS กัน ทำให้โปรแกรมที่เขียนขึ้นนี้สั้นและเข้าใจได้ง่าย

โปรแกรมที่ยกตัวอย่างนี้ จะใช้ไฟล์ NAMECARD ใน ALIAS card คือ select 1 และ Select 2 สำหรับ ALIAS ชื่อ borr จากนั้นจะเซตความสัมพันธ์ไว้

โปรแกรมจะลูปผ่าน DO WHILE เพื่อค้นหาข้อมูล การค้นหาข้อมูลจะถามชื่อใส่ในตัวแปร MNAME หลังจากนั้นจะ locate หาตำแหน่งโดยใช้คำสั่ง locate คำสั่งนี้จะค้นหาข้อมูลตามที่กำหนด ถ้าพบจะหยุดเรคอร์ดพอยเตอร์ไว้ ถ้าไม่พบจะเลื่อนไปเรื่อยๆจนจบไฟล์และจะได้สถานะ EOF( ) เป็นจริง เราจึงใช้ IF EOF( ) เพื่อตรวจสอบว่าค้นพบหรือไม่ได้

6

```
CLEAR ALL
SET TALK OFF
SELECT 2
USE BORROW INDEX BORROW ALIAS BORR
SELECT 1
USE NAMECARD ALIAS CARD
set relation to idno into borr
ACTION = "Y"
DO WHILE UPPER(action)="Y"
   CLEAR
   store "                " to mname
   @ 5,20 SAY "File Relation ..Two files"
   @ 7,20 say "      dBASE Corner"
   @ 9,20 say "        MICRO. 22 "
   @ 15,10 say "Enter name..." get mname
   read
   go top
   locate for name =TRIM(mname)
   @ 17,0 clear
   IF eof()
      ? "Data not found"
   ELSE
      ? "idno =",idno,"  name =",name
      ? "address = ",address, "   telephone = ",telephone
      ?
      select borr
      do while idno = card->idno
         ? "bookname = ",bookname," dateborrow = ",dateborrow," status = ",statu
s
         skip
      enddo
      select card
   ENDIF
   ?
   accept "Do you want to find another person..Y/N?" to action
enddo
clear all
return
```

ในกรณีที่ค้นพบ เรคอร์ดพอยเตอร์ของ ALIAS borr จะเลื่อนมาตรงกับ IDNO ที่ต้องการ ดังนั้นเราจึงสามารถขอดูข้อมูลได้ แต่เนื่องจากอาจมีหลายเรคอร์ดที่ IDNO เหมือนกัน จึงใช้ลูป DO WHILE ซ้อนอีกครั้ง

โปรแกรมที่เขียนเป็นดังรูปที่ 6

โปรแกรมสำหรับค้นหาข้อมูลจาก 2 ไฟล์ โดยใช้ชื่อเป็นการค้นหานี้ ใช้หลักการง่ายๆ ที่ทำได้ โดยให้ไฟล์แรกค้นแบบลำดับด้วย locate ไฟล์ที่ 2 ค้นหาแบบ index โดยใช้ relation ของดีเบส

เมื่อให้โปรแกรมทำงานจะปรากฏภาพดังรูปที่ 7

7

```
                File Relation ..Two files

                dBASE Corner

                MICRO. 22



          Enter name... johny

idno = A002   name = johny walker
address =  2 sukhumvit 29 Bkk.      telephone =  3424566

bookname =  Thaieasy ver4.5  dateborrow =  04/08/86  status =  .F.
bookname =  lotus release 2  dateborrow =  04/25/86  status =  .T.
bookname =  C programming    dateborrow =  02/26/86  status =  .F.

Do you want to find another person..Y/N?
```

เมื่อรันโปรแกรม ดีเบสจะถามชื่อ หาก
ชื่อลงไป ดีเบสจะค้นไฟล์แรกให้ idno
me address และ telephone ดีเบสจะนำ
no ไป relation ตามที่ตั้งไป หา book-
me, dateborrow และ statur มาให้ ในที่นี้
3 เรคอร์ด

เมื่อต้องการรันลูปอีกให้ตอบ Y ผลที่
จะเหมือนเดิมดังรูปที่ 8

คราวนี้ใส่ชื่อ dangdum ผลที่ได้จะค้น
และได้ข้อมูลถูกต้อง แต่หากใส่ชื่อที่ไม่มี
แฟ้มข้อมูล ผลที่เกิดขึ้นจะเป็นดังรูปที่ 9

รป้อนข้อมูลในฐานข้อมูล
ลายแฟ้มข้อมูล)

ฐานข้อมูลที่ออกแบบไว้จะประกอบ
ย แฟ้มข้อมูลจำนวนมากหลายแฟ้มที่เชื่อม
สัมพันธ์กัน การป้อนข้อมูลจะต้องให้ข้อมูล
เข้าไปยังแฟ้มข้อมูลต่างๆ ได้ถูกต้อง ในที่นี้
จึงขอสมมติการป้อนข้อมูลในไฟล์ข้อมูลที่
เราสมมติขึ้นมาคือ NAMECARD และ
BORROW โดยเมื่อป้อนชื่อก่อน จากนั้น
จะนำชื่อไปเปรียบเทียบกับข้อมูลในไฟล์
NAMECARD ถ้ามีอยู่แล้วจะให้ป้อนชื่อ
หนังสือที่ยืม แต่ถ้ายังไม่มีจะให้ป้อนราย
ละเอียดเกี่ยวกับ NAMECARD ก่อน

โดยเริ่มต้นบนจอภาพจะเป็นดังรูปที่ 10

8

```
File Relation ..Two files
dBASE Corner
MICRO. 22

Enter name... dangdum

idno = A003    name = dangdum seesod
address =  2 vipavadee Bkk.          telephone =  5438765
bookname =  8086 Assembly    dateborrow =  04/17/86  status =  .T.
Do you want to find another person..Y/N?
```

9

```
File Relation ..Two files
dBASE Corner
MICRO. 22

Enter name... nuanprang

Data not found
Do you want to find another person..Y/N?
```

10

```
File relation ..TWO file
dBASE corner
MICRO 22

Enter name...
```

11

```
File relation ..TWO file

dBASE corner

MICRO 22

Enter name... sodsai vilaita                NEW MEMBER

Last idno =A005
You should add one for next id no

Enter Id no .. A006
Enter name..sodsai vilaita
Enter Address.. 1 bangbua Bkk.
Enter Telephone no... 5791126

Enter bookname.. lotus 123
Enter dateborrow.. 04/23/86
Enter Status.. t
do you want to add more data y/n?
```

12

```
File relation ..TWO file

dBASE corner

MICRO 22

Enter name... dangdum

Enter Id no ..A003
Enter name..dangdum seesod
Enter Address..2 vipavadee Bkk.
Enter Telephone no...5438765

Enter bookname.. dcorner micro22
Enter dateborrow.. 03/26/86
Enter Status.. t
do you want to add more data y/n?
```

ครั้นเมื่อป้อนชื่อ เช่น sodsai vilaita จะปรากฏว่า NEW MEMBER จากนั้นจะบอกว่า ID สุดท้ายที่ใช้แล้วคืออะไร ให้บวกหนึ่งแล้วป้อนต่อไป โดยส่วนที่ขีดเส้นใต้คือ การป้อนเข้าไป ดังรูปที่ 11

แต่ถ้าหากว่าเราป้อนข้อมูลที่มีอยู่แล้ว เช่น ชื่อ dangdum seesod ดีเบสจะให้ป้อนเฉพาะรายการของหนังสือที่ยืมเท่านั้น และจะไม่มีคำว่า NEW MEMBER ดังรูปที่ 12

เมื่อป้อนข้อมูลเสร็จแล้ว ลอง list ข้อมูลทั้ง 2 ไฟล์ดู จะเห็นว่าข้อมูลเข้าไปยังตำแหน่งที่ต้องการถูกต้อง ดังรูปที่ 13

13

```
use namecard
. list
Record#  IDNO NAME                 ADDRESS              TELEPHONE
      1  A001 srisuda varaporn     1 new road Bkk.      2324322
      2  A002 johny walker         2 sukhumvit 29 Bkk.  3424566
      3  A003 dangdum seesod       2 vipavadee Bkk.     5438765
      4  A004 ngamta sopapan       3 paholyothin        5792534
      5  A005 varaporn vorawan     2 ngamvongwan Bkk.   5790777
      6  A006 sodsai vilaita       1 bangbua Bkk.       5791126
. use borrow index borrow
. list
Record#  IDNO BOOKNAME          DATEBORROW STATUS
      1  A001 dBASE III         03/21/86   .F.
      2  A001 lotus 123         04/11/86   .F.
      3  A002 Thaieasy ver4.5   04/08/86   .F.
      4  A002 lotus release 2   04/25/86   .T.
      5  A002 C programming     02/26/86   .F.
     10  A002 symphony          04/23/86   .T.
      6  A003 8086 Assembly     04/17/86   .T.
     12  A003 multiplan         04/14/86   .T.
     14  A003 dcorner micro22   03/26/86   .T.
      7  A004 Local area net.   04/25/86   .F.
      8  A004 Thaieasy ver4.1   02/24/86   .F.
      9  A004 MSDOS             04/21/86   .T.
     11  A005 Thaiword          05/19/86   .F.
     13  A006 lotus 123         04/23/86   .T.
```

14

```
*program data entry for two relation file
*
clear all
set talk off
clear
select 4
use borrow index borrow alias borr
select 1
use namecard alias card
store "Y" to action
@ 5,20 say "File relation ..TWO file"
@ 7,20 say "        dBASE corner"
@ 9,20 say "         MICRO 22"
do while upper(action)= "Y"
   @ 10,0 clear
   store "                          " to mname
   @ 11,10 say "Enter name..." get mname
   read
   go top
   locate for name = trim(mname)
   if eof()
      @ 11,50 say "NEW MEMBER"
      go bottom
      @ 13,10 say "Last idno =" + idno
      @ 14,10 say "You should add one for next id no"
      appe blank
      @ 16,10 say "Enter Id no .." get idno
      replace name with trim(mname)
      @ 17,10 say "Enter name.." + name
      @ 18,10 say "Enter Address.." get address
      @ 19,10 say "Enter Telephone no..." get telephone
      read
      select borr
      appe blank
      replace idno with card->idno
      @ 21,10 say "Enter bookname.." get bookname
      @ 22,10 say "Enter dateborrow.." get dateborrow
      @ 23,10 say "Enter Status.." get status
      read
      select card
  else
      @ 16,10 say "Enter Id no .." + idno
      @ 17,10 say "Enter name.." + name
      @ 18,10 say "Enter Address.." + address
      @ 19,10 say "Enter Telephone no..." + telephone
      select borr
      appe blank
      replace idno with card->idno
      @ 21,10 say "Enter bookname.." get bookname
      @ 22,10 say "Enter dateborrow.." get dateborrow
      @ 23,10 say "Enter Status.." get status
      read
      select card
   endif
   accept "do you want to add more data y/n?" to action
enddo
clear all
return
*
```

โปรแกรมที่ใช้ในการจัดการการป้อนข้อมูลเป็นดังรูปที่ 14

โกลบอลและโลคอล

ดีเบสทรีนำหลักการของภาษาคอมพิวเตอร์สมัยใหม่มาใช้ ซึ่งแตกต่างจากดีเบสทูไปมาก กล่าวคือในแต่ละโปรแกรมจะถือว่าตัวแปรที่ใช้เป็นตัวแปรท้องถิ่นหรือโลคอล ดังนั้นเมื่อออกจากโปรแกรมจะทำการ release ตัวแปรหมดโดยไม่ต้องใช้คำสั่ง release โดยโครงสร้างของตัวแปร สามารถกำหนดให้เป็นได้ทั้งแบบโกลบอลและโลคอล ทำให้การใช้ตัวแปรไม่สับสน และง่ายต่อการตรวจสอบ เพราะตัวแปรแต่ละตัวจะเป็นตัวแปรท้องถิ่น

ลองดูความหมายของการกำหนดตัวแปร ดีเบสทรีกำหนดตัวแปรท้องถิ่นด้วยคำสั่ง private (ไม่กำหนดก็ถือว่าท้องถิ่นอยู่แล้ว) และใช้คำสั่ง public เป็นการกำหนดตัวแปรโกลบอล

15

```
. var1=5
. var2=8
. var3=var1+var2
. disp memo
VAR1            pub    N             5  (          5.00000000)
VAR2            pub    N             8  (          8.00000000)
VAR3            pub    N            13  (         13.00000000)
     3 variables defined,       27 bytes used
   253 variables available,   5973 bytes available
. clear all
. disp memo
     0 variables defined,        0 bytes used
   256 variables available,   6000 bytes available
.
```

dBASE

พิจารณาจากการทำงานดังรูปที่ 15

ถ้ากำหนดตัวแปร var1 = 5, var2 = 0, var3 = var1 + var2 และขอดูตัวแปรด้วยคำสั่ง disp memo จะได้ผลดังกล่าวแล้ว

หากใช้คำสั่ง clear all จะเป็นการ clear ทุกสิ่งทุกอย่าง เช่น ปิดไฟล์ ยกเลิกตัวแปรทั้งหมด

คราวนี้สมมติสร้างโปรแกรมหนึ่งโปรแกรมดังต่อไปนี้

```
*Programm for MICRO 22
*
set talk off
clear all
a = 5
b = 8
c = a+b
? "a + b =",c
return
```

โปรแกรมนี้สร้างตัวแปรไว้ 3 ตัว คือ a,b,c แต่เมื่อเราลองรันโปรแกรมนี้ดู เสร็จแล้วขอดูว่ามีตัวแปรเก็บหรือไม่ ด้วยคำสั่ง disp memo ปรากฏว่าไม่มี ดังนั้นจะเห็นได้ว่า ตัวแปรที่สร้างในแต่ละโปรแกรมจะเป็นตัวแปรท้องถิ่น ไม่สามารถนำมาใช้ข้ามกันได้ ดังรูปที่ 16

คราวนี้ลองมาทำการสร้างโปรแกรมชื่อ test โดยกำหนดตัวแปรเป็นแบบ public หรือโกลบอล เพื่อให้โปรแกรมอื่นเรียกใช้ได้ การกำหนดจะต้องบอกว่า จะให้ตัวแปรใดบ้างเป็นโกลบอล โดยใส่เป็นลิสต์ไว้หลังคำสั่ง public ดังนี้

```
. type test.prg
*public
set talk off
public a,b,c
a = 123
b = 456
c = a+b
? c
return
```

โปรแกรมนี้สร้างตัวแปรไว้ คือ a,b,c เป็นแบบโกลบอล ดังนั้นหากรันโปรแกรมนี้แล้ว ตัวแปรนี้จะยังคงอยู่ในหน่วยความจำ เพื่อให้โปรแกรมอื่นเรียกใช้ได้ ดังรูปที่ 17

16

```
do prog3
Do you want some help? (Y/N) No
. do pro3
a + b =          13
.
. disp memo
    0 variables defined,          0 bytes used
  256 variables available,    6000 bytes available
.
```

17

```
disp memo
    0 variables defined,          0 bytes used
  256 variables available,    6000 bytes available
. do test
      579
. disp memo
A               pub    N          123   (         123.00000000)
B               pub    N          456   (         456.00000000)
C               pub    N          579   (         579.00000000)
    3 variables defined,         27 bytes used
  253 variables available,     5973 bytes available
```

ในทำนองเดียวกัน เราสามารถใช้คำสั่ง private เพื่อกำหนดให้เป็นตัวแปรแบบท้องถิ่น ทำให้เราไม่ต้องพะวงเรื่องชื่อซ้ำ เพราะชื่อในแต่ละโปรแกรม จะซ้ำกันได้โดยเห็นเป็นคนละตัว โดยลองพิจารณาจากโปรแกรมดังนี้

```
. type pro4.prg
*private variable
*
set talk off
private m,n,o
m=5
input "enter for n=" to n
o =123
return
```

ตัวแปร m,n,o ในที่นี้จะเป็นโลคอล คือใช้เฉพาะในโปรแกรมนี้เท่านั้น ถ้าม m,n,o ในโปรแกรมอื่น จะเห็นเป็นตัวแปรไ

ในที่นี้จะลองสร้างโปรแกรมอีกโปร แกรมเพื่อเรียก pro4 ดู โดยใช้ตัวแปรช เดียวกัน ดังนี้

```
. type pro5.prg
*test for local variable
*
m=1
n=2
o=3
do pro4
? "m=",m
? "n=",n
? "o=",o
return
```

ในที่นี้กำหนดให้ m,n,o มีค่าก่ แล้วจึงเรียก pro4 เสร็จแล้วเรานำเอาค่า n,o มาพิมพ์จะพบว่าได้ค่าเดิม กล่าวคือ n,o ต่างโปรแกรมกัน จะเห็นเสมือนเป็นแปรคนละตัวกันดังนี้

```
. do pro5
enter for n=678
m=              1
n=              2
o=              3
```

การตั้งกระบวนความ (procedure)

ในภาษาปาสคาลมีการสร้างคำใหม่ที่เรียกว่ากระบวนความ การใช้กระบวนความจึงเป็นเรื่องที่สำคัญของโปรแกรมคอมพิวเตอร์สมัยใหม่ ดีเบสทรีมีลักษณะการตั้งกระบวนความ ได้เหมือนภาษาคอมพิวเตอร์สมัยใหม่ทั่วไป โดยแต่ละกระบวนความ ก็คล้ายกับเป็นโปรแกรมย่อยที่จะเรียกใช้ได้ทุกเมื่อ แต่เนื่องจากในกระบวนความ จะใช้เป็นตัวแปรท้องถิ่นที่เรียกว่าโลคอล ดังนั้นจึงต้องมีวิธีการผ่านค่าให้กับตัวแปรเหล่านั้น เทคนิคการผ่านค่านี้เรียกว่า การผ่านพารามิเตอร์ให้

ข้อดีของกระบวนความในดีเบสทรีซึ่งเด่นชัดจากดีเบสทูมากๆ ก็คือ จากโปรแกรมหลักสามารถเซทกระบวนความไปยังไฟล์หนึ่ง และในไฟล์ที่เป็นกระบวนความนี้ อาจจะมีหลายสิบหลายร้อยกระบวนความ (เหมือนโปรแกรมย่อย) ดังนั้นการเปิดไฟล์.PRGจึงใช้เพียงไฟล์โปรแกรมหลัก และไฟล์กระบวนความเท่านั้นได้ จึงไม่จำเป็นต้องให้แต่ละโปรแกรมย่อยเป็นไฟล์ๆ หนึ่งอีกต่อไป

ลองดูการสร้างไฟล์กระบวนความจากตัวอย่างตามรูปที่ 18 โดยตัวไฟล์นี้คือ proc1.prg ซึ่งเป็นการสร้างไฟล์เหมือน command file อื่นทั่วไป อีกทั้งชนิดของไฟล์ก็เป็น .prg เหมือนกัน กระบวนความในไฟล์นี้จึงมีได้หลายกระบวนความ และแต่ละกระบวนความจะเรียกตัวกลับไปมาอย่างไรก็ได้

ในแต่ละกระบวนความ ถ้าจะให้มีการส่งผ่านค่าตัวแปรกัน เราจะใช้คำสั่งว่า parameter <ตัวแปร,....> และถ้าต้องการกำหนดตัวแปรใดเป็นตัวแปรโกลบอลก็ทำได้

18

```
. type proc1.prg
PROCEDURE one
clear
@ 15,30 say "Chaiyoo...procedure 1"
@ 16,20 say "      OK.."
return
*
*
PROCEDURE two
PARAMETER varstring,num
clear
@ 15,20 say varstring
?
? "           and number = ",num
return
**
**
PROCEDURE three
PARAMETER st1,st2,num1
PUBLIC x,y,z
clear
@ 10,10 say st1
@ 14,10 say st2
? "         number for parameter = ",num1
wait
x = "dang"
y = "dum"
z = "nidnoy"
do two with x,5
return
***
***
PROCEDURE four
clear
@ 10,10 say "Now we are in procedure four"
?
wait
do three with "Goodmorning","Goodnight",15
?
? "Hear return to procedure four"
? "variable x = ",x
? "variable y = ",y
? "variable z = ",z
return
```

19

```
. type proc2.prg
*Main program call procedure
set procedure to proc1
do one
wait
a = "Hello"
b = 29
c = "main"
do two with a,b
wait
do three with "Thai Co.",a,b
wait
do four
? "variable a=",a
? "variable b =",b
? "variable c = ",c
?
?
? "variable x = ",x
? "variable y = ",y
? "variable z = ",z
wait
return
```

ลองดูการเรียกใช้โดยใช้ไฟล์หลักเป็นไฟล์ที่จะแสดงดังในรูปที่ 19 สังเกตว่าเรามีคำสั่ง Set procedure to <ชื่อไฟล์กระบวนความ> หลังจากนั้นจะเรียกกระบวนความอะไรก็ได้ ด้วยการใช้ do <ชื่อกระบวนความ>

อนึ่งโดยปกติหากเลิกใช้ไฟล์กระบวนความ จะต้องปิดไฟล์นี้ด้วย มิฉะนั้นหากเราจะเรียกอีก จะเกิดข้อผิดพลาดว่าไฟล์ได้เปิดแล้ว การปิดเราใช้ close procedure ก็เป็นอันเสร็จการปิดไฟล์